Chinese Journal of Aeronautics, (2016), 29(1): 215–227

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

Indirect Radau pseudospectral method for the receding horizon control problem

Yuxin Liao [a,*], Huifeng Li [a], Weimin Bao [a,b]

[a] School of Astronautics, Beihang University, Beijing 100083, China
[b] Science and Technology Committee, China Aerospace Science and Technology Corporation, Beijing 100048, China

Received 15 May 2015; revised 22 July 2015; accepted 13 October 2015
Available online 14 January 2016

KEYWORDS

Indirect Radau pseudospectral method;
Linear time-varying system;
Nonlinear system;
Receding horizon control;
Two-point boundary value problem

Abstract To solve the receding horizon control (RHC) problem in an online manner, a novel numerical method called the indirect Radau pseudospectral method (IRPM) is proposed in this paper. Based on calculus of variations and the first-order necessary optimality condition, the RHC problem for linear time-varying (LTV) system is transformed into the two-point boundary value problem (TPBVP). The Radau pseudospectral approximation is employed to discretize the TPBVP into well-posed linear algebraic equations. The resulting linear algebraic equations are solved via a matrix partitioning approach afterwards to obtain the optimal feedback control law. For the nonlinear system, the linearization method or the quasi linearization method is employed to approximate the RHC problem with successive linear approximations. Subsequently, each linear problem is solved via the similar method which is used to solve the RHC problem for LTV system. Simulation results of three examples show that the IRPM is of high accuracy and of high computation efficiency to solve the RHC problem and the stability of closed-loop systems is guaranteed.

1. Introduction

Receding horizon control, also known as model predictive control, originates from process industry in the early 1970s,[1] and has been successfully applied in different fields such as aerospace engineering,[1–8] environmental sciences,[9] chemical engineering[10] and robotics.[11,12] The basic idea of receding horizon control (RHC) is that, an optimal control problem based on current states of a dynamic system is repeatedly solved online over a moving horizon, while only the first sequence of the obtained optimal control laws is implemented as the current control input. Since the time interval of the performance index for the RHC problem is finite, the optimal feedback control law can be determined even for a system which is not open-loop stable.[13]

In most cases, it is difficult to find analytical solutions for RHC problems, especially for time-varying systems. Traditional methods[13,14] rely on solving the two-point boundary value problem (TPBVP) resulting from the RHC problem by backward sweep of the Riccati differential equation or by transition matrices. It is well-known that these methods are not

* Corresponding author. Tel.: +86 10 82339753.
E-mail addresses: liaoyuxin@buaa.edu.cn (Y. Liao), lihuifeng@buaa.edu.cn (H. Li).

Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.cja.2015.12.023

suitable for online implementation of RHC since they are numerically unstable and time consuming.[7,15–18] To address the above issue, many numerical methods[6,15–18] have been proposed. By employing standard numerical integration and quadrature formulas to approximate the state derivatives and the integral cost, Lu[15] transformed the RHC problem for linear time-varying (LTV) system into a quadratic programming (QP) problem and then obtained approximate closed-form time-varying control laws by solving the resulting QP problem analytically. Yan,[16] Williams[17] and Yang et al.[6] proposed the indirect Legendre pseudospectral method, the indirect Jacobi pseudospectral method, and the indirect Gauss pseudospectral method to solve the RHC problem, respectively. The core idea of the preceding methods was that the corresponding pseudospectral approximations were utilized to transform the TPBVP resulting from the RHC problem into a set of linear algebraic equations. With the variational principle and the generating function, Peng et al.[18] proposed an efficient sparse approach which transforms the resulting TPBVP into a set of sparse symmetric positive definite linear algebraic equations. As linear algebraic equations could be efficiently and accurately solved, both the indirect pseudospectral methods[6,16,17] and Peng's approach[18] were highly effective for online implementation of RHC.

In this paper, motivated by Refs.[6,16,17], we propose a novel numerical method called the indirect Radau pseudospectral method to solve the RHC problem in an online manner. The proposed method utilizes the Radau pseudospectral approximation to transform the TPBVP, which results from the RHC problem for LTV system based on calculus of variations and the first-order necessary optimality condition, into well-posed linear algebraic equations. The optimal feedback control law at the current time is then obtained by solving the resulting linear algebraic equations via a matrix partitioning approach. For the nonlinear system, successive linear approximations are obtained through the linearization method or the quasi linearization method. Then all that remains is to solve the resulting linear problems using the similar method for solving the RHC problem for LTV system. Since the collocation points of each pseudospectral approximation are distinguishing, the indirect Radau pseudospectral method (IRPM) and other indirect pseudospectral methods[6,16,17] employing different pseudospectral approximations transform the resulting TPBVP into linear algebraic equations with different forms and dimensions. Moreover, the boundary conditions of TPBVP are incorporated in different ways in each method. The effectiveness of the proposed method is validated by three examples. The first one is an open-loop unstable LTV system with an analytical model. The second one is the single inverted pendulum which is a nonlinear system. The third one is the space shuttle reentry guidance problem which can be viewed as a trajectory tracking problem for nonlinear system. Simulation results demonstrate that the IRPM can solve the RHC problem online with high accuracy and high computation efficiency and the closed-loop systems are stable with the resulting optimal feedback control law.

Recently, the Radau pseudospectral method (RPM),[19–22] as a direct collocation method for numerically solving optimal control problems, has attracted wide attention. The RPM transforms the optimal control problem into the nonlinear programming problem which is then solved by numerical optimization methods such as the sequential quadratic programming method and the interior point method, whereas the IRPM transforms the TPBVP resulting from the RHC problem into a set of linear algebraic equations which is then solved efficiently and accurately by a matrix partitioning approach. The only similarity of the IRPM proposed in this paper and the RPM[19–22] is that the Radau pseudospectral approximation is employed by both methods to approximate the derivatives of corresponding variables.

2. Problem formulation

2.1. RHC problem for LTV system

Consider a LTV system expressed as

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(\tau)\boldsymbol{x} + \boldsymbol{B}(\tau)\boldsymbol{u} \\ \boldsymbol{x}(\tau_0) = \boldsymbol{x}_0 \end{cases} \tag{1}$$

where $\boldsymbol{x} \in \mathbf{R}^n$ and $\boldsymbol{u} \in \mathbf{R}^m$ are state and control variables, respectively; $\boldsymbol{A}(\tau) \in \mathbf{R}^{n\times n}$ and $\boldsymbol{B}(\tau) \in \mathbf{R}^{n\times m}$ are continuous coefficient matrices; $\boldsymbol{x}_0$ is the initial condition. The system is assumed to be uniformly completely controllable.[15]

The RHC problem for LTV system at any fixed time $t \geqslant \tau_0$, as described in Refs.[15,18], is defined to be an optimal control problem in which the performance index

$$J = \frac{1}{2}\int_t^{t+T} \left(\boldsymbol{x}^{\mathrm{T}}(\tau)\boldsymbol{Q}(\tau)\boldsymbol{x}(\tau) + \boldsymbol{u}^{\mathrm{T}}(\tau)\boldsymbol{N}(\tau)\boldsymbol{u}(\tau)\right)\mathrm{d}\tau \tag{2}$$

is minimized for some chosen $\delta \leqslant T \leqslant \infty$ (T is the horizon length and δ is a positive constant), the nonnegative definite symmetric weighted matrix $\boldsymbol{Q} \in \mathbf{R}^{n\times n}$ and the positive definite symmetric weighted matrix $\boldsymbol{N} \in \mathbf{R}^{m\times m}$, subject to the LTV system dynamics with the initial condition shown as Eq. (1) and the constraint

$$\boldsymbol{x}(t+T) = \boldsymbol{0} \tag{3}$$

The goal of RHC for LTV system is to solve the optimal control $\boldsymbol{u}_{\mathrm{optm}}(\tau)$ for the above problem at the time interval $\tau \in [t, t+T]$ with the current state $\boldsymbol{x}(t)$ as the initial condition, where only the first data $\boldsymbol{u}_{\mathrm{optm}}(t)$ is used as the current control input $\boldsymbol{u}(t)$ to the system, and the rest of $\boldsymbol{u}_{\mathrm{optm}}(\tau)$ is discarded. For the next instantaneous time t, the preceding solution process is repeated and the control input is recomputed. According to Ref.[23], the control input obtained by such an RHC strategy is given by

$$\boldsymbol{u}(t) = -\boldsymbol{N}^{-1}(t)\boldsymbol{B}^{\mathrm{T}}(t)\boldsymbol{P}^{-1}(t, t+T)\boldsymbol{x}(t) \tag{4}$$

where $\boldsymbol{P}(t, t+T) \in \mathbf{R}^{n\times n}$ satisfies the matrix differential Riccati equation (MDRE)

$$\begin{aligned} \dot{\boldsymbol{P}}(\tau, t_T) = {} & \boldsymbol{A}(\tau)\boldsymbol{P}(\tau, t_T) + \boldsymbol{P}(\tau, t_T)\boldsymbol{A}^{\mathrm{T}}(\tau) \\ & + \boldsymbol{P}(\tau, t_T)\boldsymbol{Q}(\tau)\boldsymbol{P}(\tau, t_T) - \boldsymbol{B}(\tau)\boldsymbol{N}^{-1}(\tau)\boldsymbol{B}^{\mathrm{T}}(\tau) \end{aligned} \tag{5}$$

at any time $\tau \in [t, t+T] \Longleftrightarrow [t, t_T]$ with the terminal boundary condition

$$\boldsymbol{P}(t_T, t_T) = \boldsymbol{P}(t+T, t+T) = \boldsymbol{0}_{n\times n} \tag{6}$$

To get the current control input $\boldsymbol{u}(t)$ from Eq. (4), the MDRE shown as Eq. (5) needs to be integrated backward from $t+T$ to t for every current time t with the boundary condition shown as Eq. (6). As noted above, online integration of the MDRE for every $t \geqslant \tau_0$ is a time-consuming work and

may pose numerical unstability. Therefore, to avoid the integration of MDRE and improve the efficiency in real-time computation of RHC for LTV system, the IRPM is developed to transform the above problem into linear algebraic equations which can be efficiently and accurately solved online.

2.2. RHC problem for nonlinear system

Consider a nonlinear system with the initial condition expressed as

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau) \\ \boldsymbol{x}(\tau_0) = \boldsymbol{x}_0 \end{cases} \tag{7}$$

where $\boldsymbol{f}$ is a function vector in which all elements are continuous and differentiable. The RHC problem for nonlinear system at any fixed time $t \geqslant \tau_0$, as described in Ref.[17], is defined to find the optimal control $\boldsymbol{u}_{\text{optm}}(\tau)$ that minimizes the performance index

$$\begin{aligned} \tilde{J} = & \frac{1}{2}(\boldsymbol{M}_{\text{f}}\boldsymbol{x}(t+T) - \boldsymbol{\psi})^{\text{T}}\boldsymbol{S}_{\text{f}}(\boldsymbol{M}_{\text{f}}\boldsymbol{x}(t+T) - \boldsymbol{\psi}) \\ & + \frac{1}{2}\int_t^{t+T} \Big[(\boldsymbol{x}(\tau) - \boldsymbol{x}_{\text{d}}(\tau))^{\text{T}}\boldsymbol{Q}(\tau)(\boldsymbol{x}(\tau) - \boldsymbol{x}_{\text{d}}(\tau)) \\ & + (\boldsymbol{u}(\tau) - \boldsymbol{u}_{\text{d}}(\tau))^{\text{T}}\boldsymbol{N}(\tau)(\boldsymbol{u}(\tau) - \boldsymbol{u}_{\text{d}}(\tau))\Big]\text{d}\tau \end{aligned} \tag{8}$$

subject to the nonlinear system dynamics and the initial condition shown as Eq. (7), where $\boldsymbol{x}_{\text{d}} \in \mathbf{R}^n$ and $\boldsymbol{u}_{\text{d}} \in \mathbf{R}^m$ are desired time-varying state and control variables, respectively; $\boldsymbol{S}_{\text{f}} \in \mathbf{R}^{n\times n}$ is the nonnegative definite symmetric weighted matrix; $\boldsymbol{\psi} \in \mathbf{R}^n$ is the desired vector of values for linear combination of desired final state variable $\boldsymbol{M}_{\text{f}}\boldsymbol{x}(t+T)$ and $\boldsymbol{M}_{\text{f}} \in \mathbf{R}^{n\times n}$ is a given matrix.

As same as the goal of RHC for LTV system mentioned in Section 2.1, the nonlinear optimal control problem defined by Eqs. (7) and (8) is solved repeatedly as time goes on, while the obtained optimal control input is only applied at the current time.

3. Indirect Radau pseudospectral method

3.1. Method for solving RHC problem for LTV system

Based on calculus of variations and the first-order necessary optimality condition, the solution of the RHC problem for LTV system should satisfy the following Hamiltonian canonical equation[18]:

$$\begin{cases} \dot{\boldsymbol{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \boldsymbol{A}(\tau)\boldsymbol{x} - \boldsymbol{B}(\tau)\boldsymbol{N}^{-1}(\tau)\boldsymbol{B}^{\text{T}}(\tau)\boldsymbol{\lambda} \\ \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{x}} = -\boldsymbol{Q}(\tau)\boldsymbol{x} - \boldsymbol{A}^{\text{T}}(\tau)\boldsymbol{\lambda} \\ \boldsymbol{x}(t) = \tilde{\boldsymbol{x}}_0 \\ \boldsymbol{x}(t+T) = \boldsymbol{0} \end{cases} \tag{9}$$

where H is the Hamiltonian function expressed as

$$H = \frac{1}{2}\left(\boldsymbol{x}^{\text{T}}\boldsymbol{Q}(\tau)\boldsymbol{x} + \boldsymbol{u}^{\text{T}}\boldsymbol{N}(\tau)\boldsymbol{u}\right) + \boldsymbol{\lambda}^{\text{T}}(\boldsymbol{A}(\tau)\boldsymbol{x} + \boldsymbol{B}(\tau)\boldsymbol{u}) \tag{10}$$

$\boldsymbol{\lambda} \in \mathbf{R}^n$ is the costate variable and $\tilde{\boldsymbol{x}}_0$ is the current state variable. The optimal control is given by

$$\boldsymbol{u}_{\text{optm}} = -\boldsymbol{N}^{-1}(\tau)\boldsymbol{B}^{\text{T}}(\tau)\boldsymbol{\lambda} \tag{11}$$

To compute the optimal control $\boldsymbol{u}_{\text{optm}}$, the Hamiltonian canonical equation shown as Eq. (9) needs to be solved first. The Hamiltonian canonical equation can be expressed in the form of state space as

$$\begin{cases} \begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}(\tau) & -\boldsymbol{B}(\tau)\boldsymbol{N}^{-1}(\tau)\boldsymbol{B}^{\text{T}}(\tau) \\ -\boldsymbol{Q}(\tau) & -\boldsymbol{A}^{\text{T}}(\tau) \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} \\ \boldsymbol{x}(t) = \tilde{\boldsymbol{x}}_0 \\ \boldsymbol{x}(t+T) = \boldsymbol{0} \end{cases} \tag{12}$$

which is the TPBVP. The IRPM is proposed to solve the above TPBVP with the detailed description given as follows.

The state and costate variables are discretized at the end point $\tilde{\tau}_{N+1} = 1$ and N Legendre–Gauss–Radau (LGR) points $\tilde{\tau}_k (k = 1, 2, \ldots, N)$ distributed on the interval $[-1, 1)$. N LGR points including the initial point $\tilde{\tau}_1 = -1$ are defined as the roots of the polynomial $P_N(\tilde{\tau}) + P_{N-1}(\tilde{\tau})$, where

$$P_N(\tilde{\tau}) = \frac{1}{2^N N!} \cdot \frac{\text{d}^N}{\text{d}\tilde{\tau}^N}\left[(\tilde{\tau}^2 - 1)^N\right] \tag{13}$$

is the Nth degree Legendre orthogonal polynomial. Since the discrete points lie in the interval [−1,1], the TPBVP shown as Eq. (12) is transformed to this interval by the following linear transformation for time $\tilde{\tau} \in [\tilde{\tau}_1, \tilde{\tau}_{N+1}] = [-1, 1]$:

$$\tilde{\tau} = \frac{2(\tau - t)}{T} - 1 \tag{14}$$

It follows that the TPBVP shown as Eq. (12) can be replaced by

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \frac{T}{2}\begin{bmatrix} \boldsymbol{A}(\tilde{\tau}) & -\boldsymbol{B}(\tilde{\tau})\boldsymbol{N}^{-1}(\tilde{\tau})\boldsymbol{B}^{\text{T}}(\tilde{\tau}) \\ -\boldsymbol{Q}(\tilde{\tau}) & -\boldsymbol{A}^{\text{T}}(\tilde{\tau}) \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} \tag{15}$$

$$\begin{cases} \boldsymbol{x}(\tilde{\tau}_1) = \tilde{\boldsymbol{x}}_0 \\ \boldsymbol{x}(\tilde{\tau}_{N+1}) = \boldsymbol{0} \end{cases} \tag{16}$$

The state and costate variables are approximated as Nth degree polynomials through Lagrange interpolation as

$$\boldsymbol{x}(\tilde{\tau}) = \sum_{j=1}^{N+1} \ell_j(\tilde{\tau})\boldsymbol{x}(\tilde{\tau}_j) \tag{17}$$

$$\boldsymbol{\lambda}(\tilde{\tau}) = \sum_{j=1}^{N+1} \ell_j(\tilde{\tau})\boldsymbol{\lambda}(\tilde{\tau}_j) \tag{18}$$

where for $j = 1, 2, \ldots, N+1$,

$$\ell_j(\tilde{\tau}) = \prod_{\substack{l=1 \\ l \neq j}}^{N+1} \frac{\tilde{\tau} - \tilde{\tau}_l}{\tilde{\tau}_j - \tilde{\tau}_l} \tag{19}$$

are Nth degree Lagrange interpolation basis functions; $\tilde{\tau}_j$ are $N+1$ interpolation points which are equal to the $N+1$ discrete points.

The derivatives of $\boldsymbol{x}(\tilde{\tau})$ and $\boldsymbol{\lambda}(\tilde{\tau})$ at N LGR points are given by

$$\dot{\boldsymbol{x}}(\tilde{\tau}_k) = \sum_{j=1}^{N+1} \frac{\partial \ell_j(\tilde{\tau}_k)}{\partial \tilde{\tau}}\boldsymbol{x}(\tilde{\tau}_j) = \sum_{j=1}^{N+1} D_{kj}\boldsymbol{x}(\tilde{\tau}_j) \tag{20}$$

$$\dot{\boldsymbol{\lambda}}(\tilde{\tau}_k) = \sum_{j=1}^{N+1} \frac{\partial \ell_j(\tilde{\tau}_k)}{\partial \tilde{\tau}}\boldsymbol{\lambda}(\tilde{\tau}_j) = \sum_{j=1}^{N+1} D_{kj}\boldsymbol{\lambda}(\tilde{\tau}_j) \tag{21}$$

where D_{kj} shown as Eq. (22) are elements of the differential matrix $\boldsymbol{D} \in \mathbf{R}^{N\times(N+1)}$.

$$\begin{cases} D_{kj} = \begin{cases} \dfrac{\dot{g}(\tilde{\tau}_k)}{(\tilde{\tau}_k - \tilde{\tau}_l)\dot{g}(\tilde{\tau}_j)} & k \neq j \\ \dfrac{\ddot{g}(\tilde{\tau}_j)}{2\dot{g}(\tilde{\tau}_j)} & k = j \end{cases} \\ g(\tilde{\tau}) = (\tilde{\tau}-1)(P_N(\tilde{\tau}) + P_{N-1}(\tilde{\tau})) \end{cases} \tag{22}$$

The dynamic equations shown as Eq. (15) are discretized by imposing the condition that the derivatives of state and costate approximations expressed as Eqs. (20) and (21) satisfy the differential equation exactly at the LGR points.

Thus, by substituting Eqs. (20) and (21) into Eq. (15) and collocating at N LGR points $\tilde{\tau}_k$, Eq. (15) is transformed into the following algebraic equations:

$$\frac{2}{T}\sum_{j=1}^{N+1} D_{kj}\boldsymbol{x}_j - (\boldsymbol{A}_k\boldsymbol{x}_k - \boldsymbol{B}_k\boldsymbol{N}_k^{-1}\boldsymbol{B}_k^{\mathrm{T}}\boldsymbol{\lambda}_k) = \boldsymbol{0} \tag{23}$$

$$\frac{2}{T}\sum_{j=1}^{N+1} D_{kj}\boldsymbol{\lambda}_j + (\boldsymbol{Q}_k\boldsymbol{x}_k + \boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{\lambda}_k) = \boldsymbol{0} \tag{24}$$

where for a generic matrix $\boldsymbol{X}(\tilde{\tau})$, the notation $\boldsymbol{X}_k$ denotes $\boldsymbol{X}(\tilde{\tau}_k), (\boldsymbol{X} = \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{N}, \boldsymbol{Q}, \boldsymbol{\lambda}, \boldsymbol{x})$. Rewriting Eqs. (23) and (24) in block matrix notation, the following equation is obtained:

$$\begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{G} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\mathrm{lin}} \\ \boldsymbol{\lambda}_{\mathrm{lin}} \end{bmatrix} = \boldsymbol{V}\boldsymbol{Z} = \boldsymbol{0} \tag{25}$$

where $\boldsymbol{E}$, $\boldsymbol{F}$, $\boldsymbol{G}$, $\boldsymbol{H} \in \mathbf{R}^{Nn\times(N+1)n}$ whose the (k, j)th blocks are $n \times n$ matrices of the following form:

$$\begin{aligned} [\boldsymbol{E}]_{kj} &= \begin{cases} \dfrac{2}{T}D_{kj}\boldsymbol{I}_{n\times n} - \boldsymbol{A}_k & k = j \\ \dfrac{2}{T}D_{kj}\boldsymbol{I}_{n\times n} & k \neq j \end{cases} \\ [\boldsymbol{F}]_{kj} &= \begin{cases} \boldsymbol{B}_k\boldsymbol{N}_k^{-1}\boldsymbol{B}_k^{\mathrm{T}} & k = j \\ \boldsymbol{0}_{n\times n} & k \neq j \end{cases} \\ [\boldsymbol{G}]_{kj} &= \begin{cases} \boldsymbol{Q}_k & k = j \\ \boldsymbol{0}_{n\times n} & k \neq j \end{cases} \\ [\boldsymbol{H}]_{kj} &= \begin{cases} \dfrac{2}{T}D_{kj}\boldsymbol{I}_{n\times n} + \boldsymbol{A}_k^{\mathrm{T}} & k = j \\ \dfrac{2}{T}D_{kj}\boldsymbol{I}_{n\times n} & k \neq j \end{cases} \end{aligned} \tag{26}$$

where $\boldsymbol{x}_{\mathrm{lin}} = [\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}, \ldots, \boldsymbol{x}_{N+1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbf{R}^{(N+1)n}$; $\boldsymbol{\lambda}_{\mathrm{lin}} = [\boldsymbol{\lambda}_1^{\mathrm{T}}, \boldsymbol{\lambda}_2^{\mathrm{T}}, \ldots, \boldsymbol{\lambda}_{N+1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbf{R}^{(N+1)n}$; $\boldsymbol{Z} = [\boldsymbol{x}_{\mathrm{lin}}^{\mathrm{T}}, \boldsymbol{\lambda}_{\mathrm{lin}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbf{R}^{(2N+2)n}$; $\boldsymbol{V} \in \mathbf{R}^{2Nn\times(2N+2)n}$.

Since the boundary conditions of state variable are known as Eq. (16), Eq. (25) can be rewritten as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{G} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\mathrm{lin}} \\ \boldsymbol{\lambda}_{\mathrm{lin}} \end{bmatrix} = \boldsymbol{V}\boldsymbol{Z} &= [\,\boldsymbol{V}_1 \quad \boldsymbol{V}_{N+1} \quad \boldsymbol{V}_{\mathrm{e}}\,] \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_{N+1} \\ \boldsymbol{Z}_{\mathrm{e}} \end{bmatrix} \\ &= \boldsymbol{V}_1\boldsymbol{x}_1 + \boldsymbol{V}_{N+1}\boldsymbol{x}_{N+1} + \boldsymbol{V}_{\mathrm{e}}\boldsymbol{Z}_{\mathrm{e}} = \boldsymbol{0} \end{aligned} \tag{27}$$

where $\boldsymbol{Z}_{\mathrm{e}} = [\boldsymbol{x}_2^{\mathrm{T}}, \boldsymbol{x}_3^{\mathrm{T}}, \ldots, \boldsymbol{x}_N^{\mathrm{T}}, \boldsymbol{\lambda}_1^{\mathrm{T}}, \boldsymbol{\lambda}_2^{\mathrm{T}}, \ldots, \boldsymbol{\lambda}_{N+1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbf{R}^{2Nn}$ includes all the unknown variables; $\boldsymbol{V}_1 = \left[[\boldsymbol{E}]_{k1}^{\mathrm{T}}, [\boldsymbol{G}]_{k1}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbf{R}^{2Nn\times n}$ and $\boldsymbol{V}_{N+1} = [[\boldsymbol{E}]_{k(N+1)}^{\mathrm{T}}, [\boldsymbol{G}]_{k(N+1)}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbf{R}^{2Nn\times n}$ are submatrices of $\boldsymbol{V}$; $\boldsymbol{V}_{\mathrm{e}} \in \mathbf{R}^{2Nn\times 2Nn}$ includes the rest parts of $\boldsymbol{V}$. Then, the well-posed linear algebraic equations expressed as Eq. (27) is solved for obtaining $\boldsymbol{Z}_{\mathrm{e}}$ as

$$\begin{aligned} \boldsymbol{Z}_{\mathrm{e}} &= -\boldsymbol{V}_{\mathrm{e}}^{-1}(\boldsymbol{V}_1\boldsymbol{x}_1 + \boldsymbol{V}_{N+1}\boldsymbol{x}_{N+1}) \\ &= -\boldsymbol{V}_{\mathrm{e}}^{-1}(\boldsymbol{V}_1\tilde{\boldsymbol{x}}_0 + \boldsymbol{V}_{N+1}\boldsymbol{0}) = -\boldsymbol{V}_{\mathrm{e}}^{-1}\boldsymbol{V}_1\tilde{\boldsymbol{x}}_0 \end{aligned} \tag{28}$$

and $\boldsymbol{Z}$ can be rewritten as

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_{N+1} \\ \boldsymbol{Z}_{\mathrm{e}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n\times n} \\ \boldsymbol{0}_{n\times n} \\ -\boldsymbol{V}_{\mathrm{e}}^{-1}\boldsymbol{V}_1 \end{bmatrix} \tilde{\boldsymbol{x}}_0 = \begin{bmatrix} \boldsymbol{I}_{n\times n} \\ \boldsymbol{0}_{n\times n} \\ \boldsymbol{W} \end{bmatrix} \tilde{\boldsymbol{x}}_0 = \begin{bmatrix} \boldsymbol{W}_{\boldsymbol{x}} \\ \boldsymbol{W}_{\boldsymbol{\lambda}} \end{bmatrix} \tilde{\boldsymbol{x}}_0 \tag{29}$$

where $\boldsymbol{W}_{\boldsymbol{x}}, \boldsymbol{W}_{\boldsymbol{\lambda}} \in \mathbf{R}^{(N+1)n\times n}$ are partitions of the matrix $[\,\boldsymbol{I}_{n\times n}, \quad \boldsymbol{0}_{n\times n}, \quad \boldsymbol{W}^{\mathrm{T}}\,]^{\mathrm{T}}$, so that $\boldsymbol{x}_j$ and $\boldsymbol{\lambda}_j$ are given by

$$\begin{cases} \boldsymbol{x}_j = (\boldsymbol{W}_{\boldsymbol{x}})_j\tilde{\boldsymbol{x}}_0 \\ \boldsymbol{\lambda}_j = (\boldsymbol{W}_{\boldsymbol{\lambda}})_j\tilde{\boldsymbol{x}}_0 \end{cases} \tag{30}$$

where $(\boldsymbol{W}_{\boldsymbol{x}})_j, (\boldsymbol{W}_{\boldsymbol{\lambda}})_j \in \mathbf{R}^{n\times n}$ are partitions of $\boldsymbol{W}_{\boldsymbol{x}}$ and $\boldsymbol{W}_{\boldsymbol{\lambda}}$, respectively. The subscript j refers to the jth discrete points. Thus, by substituting Eq. (30) into Eq. (11), the discretization for the optimal control law is obtained as

$$(\boldsymbol{u}_{\mathrm{optm}})_j = -\boldsymbol{N}_j^{-1}\boldsymbol{B}_j^{\mathrm{T}}\boldsymbol{\lambda}_j = -\boldsymbol{N}_j^{-1}\boldsymbol{B}_j^{\mathrm{T}}(\boldsymbol{W}_{\boldsymbol{\lambda}})_j\tilde{\boldsymbol{x}}_0 \tag{31}$$

It is obvious that if the current state variable $\tilde{\boldsymbol{x}}_0$ is known, state, control and costate variables at the discrete points can be solved from Eqs. (30) and (31), and the values of variables at instants of time between the discrete points can be obtained by interpolation. It should be noted that the preceding optimal control law is obtained without any complex and time-consuming computation.

3.2. Method for solving RHC problem for nonlinear system

To solve the RHC problem for nonlinear system, the linearization method or the quasi linearization method is employed to transform the preceding nonlinear optimal control problem, as described in Section 2.2, into a sequence of linear optimal control problems first.

In the linearization method, the performance index shown as Eq. (8) is expanded up to second order and the system dynamics as shown in Eq. (7) are expanded up to first order with higher order terms neglected around the desired trajectory $(\boldsymbol{x}_{\mathrm{d}}, \boldsymbol{u}_{\mathrm{d}})$. The original problem is transformed into the following linear optimal control problem.

(1) Problem 1

To minimize

$$\begin{aligned} \tilde{J}_1 = &\frac{1}{2}(\boldsymbol{M}_{\mathrm{f}}\tilde{\boldsymbol{x}}(t+T) - \boldsymbol{\psi})^{\mathrm{T}}\boldsymbol{S}_{\mathrm{f}}(\boldsymbol{M}_{\mathrm{f}}\tilde{\boldsymbol{x}}(t+T) - \boldsymbol{\psi}) \\ &+ \frac{1}{2}\int_t^{t+T} \Big[(\tilde{\boldsymbol{x}}(\tau) - \boldsymbol{x}_{\mathrm{d}}(\tau))^{\mathrm{T}}\boldsymbol{Q}(\tau)(\tilde{\boldsymbol{x}}(\tau) - \boldsymbol{x}_{\mathrm{d}}(\tau)) \\ &+ (\tilde{\boldsymbol{u}}(\tau) - \boldsymbol{u}_{\mathrm{d}}(\tau))^{\mathrm{T}}\boldsymbol{N}(\tau)(\tilde{\boldsymbol{u}}(\tau) - \boldsymbol{u}_{\mathrm{d}}(\tau))\Big]\mathrm{d}\tau \end{aligned} \tag{32}$$

subject to

$$\dot{\tilde{\boldsymbol{x}}} = \hat{\boldsymbol{A}}(\tau)\tilde{\boldsymbol{x}} + \hat{\boldsymbol{B}}(\tau)\tilde{\boldsymbol{u}}, \quad \tilde{\boldsymbol{x}}(t) = \tilde{\boldsymbol{x}}_0 \tag{33}$$

where

$$\hat{\boldsymbol{A}}(\tau)=\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau),\tau)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_{\mathrm{d}}(\tau),\boldsymbol{u}_{\mathrm{d}}(\tau)} \tag{34}$$

$$\hat{\boldsymbol{B}}(\tau)=\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau),\tau)}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}_{\mathrm{d}}(\tau),\boldsymbol{u}_{\mathrm{d}}(\tau)} \tag{35}$$

In the quasi linearization method, the performance index is expanded up to second order and the system dynamics are expanded up to first order around the nominal trajectory. The original problem is transformed into the following linear optimal control problem.

(2) Problem 2

To minimize

$$\begin{aligned}\tilde{J}_2=&\frac{1}{2}(\boldsymbol{M}_{\mathrm{f}}\tilde{\boldsymbol{x}}(t+T)-\boldsymbol{\psi})^{\mathrm{T}}\boldsymbol{S}_{\mathrm{f}}(\boldsymbol{M}_{\mathrm{f}}\tilde{\boldsymbol{x}}(t+T)-\boldsymbol{\psi})\\&+\frac{1}{2}\int_t^{t+T}\Big[(\tilde{\boldsymbol{x}}(\tau)-\boldsymbol{x}_{\mathrm{d}}(\tau))^{\mathrm{T}}\boldsymbol{Q}(\tau)(\tilde{\boldsymbol{x}}(\tau)-\boldsymbol{x}_{\mathrm{d}}(\tau))\\&+(\tilde{\boldsymbol{u}}(\tau)-\boldsymbol{u}_{\mathrm{d}}(\tau))^{\mathrm{T}}\boldsymbol{N}(\tau)(\tilde{\boldsymbol{u}}(\tau)-\boldsymbol{u}_{\mathrm{d}}(\tau))\Big]\mathrm{d}\tau\end{aligned} \tag{36}$$

subject to

$$\dot{\tilde{\boldsymbol{x}}}=\underset{\sim}{\hat{\boldsymbol{A}}}(\tau)\tilde{\boldsymbol{x}}+\underset{\sim}{\hat{\boldsymbol{B}}}(\tau)\tilde{\boldsymbol{u}}+\underset{\sim}{\boldsymbol{w}}(\tau),\quad \tilde{\boldsymbol{x}}(t)=\tilde{\boldsymbol{x}}_0 \tag{37}$$

where

$$\underset{\sim}{\hat{\boldsymbol{A}}}(\tau)=\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau),\tau)}{\partial \boldsymbol{x}}\right|_{\underset{\sim}{\boldsymbol{x}}(\tau),\underset{\sim}{\boldsymbol{u}}(\tau)} \tag{38}$$

$$\underset{\sim}{\hat{\boldsymbol{B}}}(\tau)=\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau),\tau)}{\partial \boldsymbol{u}}\right|_{\underset{\sim}{\boldsymbol{x}}(\tau),\underset{\sim}{\boldsymbol{u}}(\tau)} \tag{39}$$

$$\underset{\sim}{\boldsymbol{w}}(\tau)=\boldsymbol{f}(\underset{\sim}{\boldsymbol{x}}(\tau),\underset{\sim}{\boldsymbol{u}}(\tau),\tau)-(\underset{\sim}{\hat{\boldsymbol{A}}}(\tau)\,\underset{\sim}{\boldsymbol{x}}+\underset{\sim}{\hat{\boldsymbol{B}}}(\tau)\,\underset{\sim}{\boldsymbol{u}}) \tag{40}$$

where $\underset{\sim}{\boldsymbol{w}}(\tau)$ are the higher order terms of the system dynamics; $\underset{\sim}{\boldsymbol{x}}(\tau)$ and $\underset{\sim}{\boldsymbol{u}}(\tau)$ are the state and control variables for the preceding iteration; $\tilde{\boldsymbol{x}}(\tau)$ and $\tilde{\boldsymbol{u}}(\tau)$ are the state and control variables for the current iteration. It should be noted that since the desired trajectory and the nominal trajectory are only related to time, the coefficient matrices of the linearized dynamics as shown in Eqs. (33) and (37) can be reformulated as pure function of time by substituting the values of the desired trajectory or the nominal trajectory into them at each time point.

Problem 1 can be regarded as the special case of Problem 2 with the higher order terms of the system dynamics for the linearization of the nonlinear system neglected and the nominal trajectory fixed to the desired trajectory. In view of generality, the following derivation process is based on Problem 2.

The Hamiltonian function for Problem 2 is expressed as

$$\begin{aligned}\tilde{H}=&\frac{1}{2}\Big[(\tilde{\boldsymbol{x}}(\tau)-\boldsymbol{x}_{\mathrm{d}}(\tau))^{\mathrm{T}}\boldsymbol{Q}(\tau)(\tilde{\boldsymbol{x}}(\tau)-\boldsymbol{x}_{\mathrm{d}}(\tau))\\&+(\tilde{\boldsymbol{u}}(\tau)-\boldsymbol{u}_{\mathrm{d}}(\tau))^{\mathrm{T}}\boldsymbol{N}(\tau)(\tilde{\boldsymbol{u}}(\tau)-\boldsymbol{u}_{\mathrm{d}}(\tau))\Big]\\&+\tilde{\boldsymbol{\lambda}}^{\mathrm{T}}(\underset{\sim}{\hat{\boldsymbol{A}}}(\tau)\tilde{\boldsymbol{x}}+\underset{\sim}{\hat{\boldsymbol{B}}}(\tau)\tilde{\boldsymbol{u}}+(\tau))\end{aligned} \tag{41}$$

where $\tilde{\boldsymbol{\lambda}}\in\mathbf{R}^n$ is the costate variable for the current iteration. Based on calculus of variations and the first-order necessary optimality condition, the costate equation, the transversality condition, and the optimal control law are obtained as

$$\dot{\tilde{\boldsymbol{\lambda}}}=-\frac{\partial \tilde{H}}{\partial \tilde{\boldsymbol{x}}}=-\boldsymbol{Q}(\tau)(\tilde{\boldsymbol{x}}(\tau)-\boldsymbol{x}_{\mathrm{d}}(\tau))-\underset{\sim}{\hat{\boldsymbol{A}}}^{\mathrm{T}}(\tau)\tilde{\boldsymbol{\lambda}}(\tau) \tag{42}$$

$$\tilde{\boldsymbol{\lambda}}(t+T)=\boldsymbol{M}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{f}}(\boldsymbol{M}_{\mathrm{f}}\tilde{\boldsymbol{x}}(t+T)-\boldsymbol{\psi}) \tag{43}$$

$$\frac{\partial \tilde{H}}{\partial \tilde{\boldsymbol{u}}}=\boldsymbol{0}\Rightarrow\tilde{\boldsymbol{u}}(\tau)=\boldsymbol{u}_{\mathrm{d}}(\tau)-\boldsymbol{N}^{-1}(\tau)\underset{\sim}{\hat{\boldsymbol{B}}}^{\mathrm{T}}(\tau)\tilde{\boldsymbol{\lambda}}(\tau) \tag{44}$$

Substituting Eq. (44) into Eq. (37) and combining it with Eq. (42), Eq. (43) and the current state variable $\tilde{\boldsymbol{x}}(t)=\tilde{\boldsymbol{x}}_0$, the TPBVP in the form of state space is expressed as

$$\begin{cases}\begin{bmatrix}\dot{\tilde{\boldsymbol{x}}}\\ \dot{\tilde{\boldsymbol{\lambda}}}\end{bmatrix}=\begin{bmatrix}\underset{\sim}{\hat{\boldsymbol{A}}}(\tau) & -\underset{\sim}{\hat{\boldsymbol{B}}}(\tau)\boldsymbol{N}^{-1}(\tau)\underset{\sim}{\hat{\boldsymbol{B}}}^{\mathrm{T}}(\tau)\\ -\boldsymbol{Q}(\tau) & -\underset{\sim}{\hat{\boldsymbol{A}}}^{\mathrm{T}}(\tau)\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{x}}\\ \tilde{\boldsymbol{\lambda}}\end{bmatrix}\\ \quad+\begin{bmatrix}\underset{\sim}{\hat{\boldsymbol{B}}}(\tau)\boldsymbol{u}_{\mathrm{d}}+\underset{\sim}{\boldsymbol{w}}(\tau)\\ \boldsymbol{Q}(\tau)\boldsymbol{x}_{\mathrm{d}}\end{bmatrix}\\ \tilde{\boldsymbol{x}}(t)=\tilde{\boldsymbol{x}}_0\\ \tilde{\boldsymbol{\lambda}}(t+T)=\boldsymbol{M}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{f}}(\boldsymbol{M}_{\mathrm{f}}\tilde{\boldsymbol{x}}(t+T)-\boldsymbol{\psi})\end{cases} \tag{45}$$

As before, the IRPM is proposed to transform the above TPBVP into linear algebraic equations. Since the processes of discretizing and collocating are similar to that in Section 3.1, the details are not described. Here, the results transformed from the dynamics equations in Eq. (45) for $k=1,2,\ldots,N$ are given by

$$\frac{2}{T}\sum_{j=1}^{N+1}D_{kj}\tilde{\boldsymbol{x}}_j-\underset{\sim}{\hat{\boldsymbol{A}}}_k\tilde{\boldsymbol{x}}_k+\underset{\sim}{\hat{\boldsymbol{B}}}_k\boldsymbol{N}_k^{-1}\underset{\sim}{\hat{\boldsymbol{B}}}_k^{\mathrm{T}}\tilde{\boldsymbol{\lambda}}_k=\underset{\sim}{\hat{\boldsymbol{B}}}_k(\boldsymbol{u}_{\mathrm{d}})_k+\underset{\sim}{\boldsymbol{w}}_k \tag{46}$$

$$\frac{2}{T}\sum_{j=1}^{N+1}D_{kj}\tilde{\boldsymbol{\lambda}}_j+\boldsymbol{Q}_k\tilde{\boldsymbol{x}}_k+\underset{\sim}{\hat{\boldsymbol{A}}}_k^{\mathrm{T}}\tilde{\boldsymbol{\lambda}}_k=\boldsymbol{Q}_k(\boldsymbol{x}_{\mathrm{d}})_k \tag{47}$$

Combining Eqs. (46) and (47) with Eq. (43), the following equation in block matrix notation is obtained:

$$\begin{bmatrix}\tilde{\boldsymbol{E}} & \tilde{\boldsymbol{F}}\\ \tilde{\boldsymbol{G}} & \tilde{\boldsymbol{H}}\\ \tilde{\boldsymbol{P}}_1 & \tilde{\boldsymbol{P}}_2\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{x}}_{\mathrm{non}}\\ \tilde{\boldsymbol{\lambda}}_{\mathrm{non}}\end{bmatrix}=\tilde{\boldsymbol{V}}\tilde{\boldsymbol{Z}}=\boldsymbol{T} \tag{48}$$

where $\tilde{\boldsymbol{E}},\tilde{\boldsymbol{F}},\tilde{\boldsymbol{G}},\tilde{\boldsymbol{H}}\in\mathbf{R}^{Nn\times(N+1)n}$ whose (k,j)th blocks are $n\times n$ matrices of the following form:

$$\begin{aligned}[\tilde{\boldsymbol{E}}]_{kj}&=\begin{cases}\frac{2}{T}D_{kj}\boldsymbol{I}_{n\times n}-\underset{\sim}{\hat{\boldsymbol{A}}}_k & k=j\\ \frac{2}{T}D_{kj}\boldsymbol{I}_{n\times n} & k\neq j\end{cases}\\ [\tilde{\boldsymbol{F}}]_{kj}&=\begin{cases}\underset{\sim}{\hat{\boldsymbol{B}}}_k\boldsymbol{N}_k^{-1}\underset{\sim}{\hat{\boldsymbol{B}}}_k^{\mathrm{T}} & k=j\\ \boldsymbol{0}_{n\times n} & k\neq j\end{cases}\\ [\tilde{\boldsymbol{G}}]_{kj}&=\begin{cases}\boldsymbol{Q}_k & k=j\\ \boldsymbol{0}_{n\times n} & k\neq j\end{cases}\\ [\tilde{\boldsymbol{H}}]_{kj}&=\begin{cases}\frac{2}{T}D_{kj}\boldsymbol{I}_{n\times n}+\underset{\sim}{\hat{\boldsymbol{A}}}_k^{\mathrm{T}} & k=j\\ \frac{2}{T}D_{kj}\boldsymbol{I}_{n\times n} & k\neq j\end{cases}\end{aligned} \tag{49}$$

$$\begin{cases}\tilde{\boldsymbol{P}}_1=[\boldsymbol{0}_{n\times n},\boldsymbol{0}_{n\times n},\ldots,\boldsymbol{0}_{n\times n},\boldsymbol{M}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{f}}\boldsymbol{M}_{\mathrm{f}}]\\ \tilde{\boldsymbol{P}}_2=[\boldsymbol{0}_{n\times n},\boldsymbol{0}_{n\times n},\ldots,\boldsymbol{0}_{n\times n},-\boldsymbol{I}_{n\times n}]\end{cases} \tag{50}$$

$$\boldsymbol{T}=\begin{bmatrix}\underset{\sim}{\hat{\boldsymbol{B}}}_k(\boldsymbol{u}_{\mathrm{d}})_k+\underset{\sim}{\boldsymbol{w}}_k\\ \boldsymbol{Q}_k(\boldsymbol{x}_{\mathrm{d}})_k\\ \boldsymbol{M}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{f}}\boldsymbol{\psi}\end{bmatrix} \tag{51}$$

where $\tilde{\boldsymbol{P}}_1, \tilde{\boldsymbol{P}}_2 \in \mathbf{R}^{n\times(N+1)n}$; $\boldsymbol{T} \in \mathbf{R}^{(2N+1)n}$; $\tilde{\boldsymbol{x}}_{\text{non}} = [\tilde{\boldsymbol{x}}_1^{\mathrm{T}}, \tilde{\boldsymbol{x}}_2^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{x}}_{N+1}^{\mathrm{T}}]^{\mathrm{T}} \in R^{(N+1)n}$; $\tilde{\boldsymbol{\lambda}}_{\text{non}} = [\tilde{\boldsymbol{\lambda}}_1^{\mathrm{T}}, \tilde{\boldsymbol{\lambda}}_2^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{\lambda}}_{N+1}^{\mathrm{T}}]^{\mathrm{T}} \in \boldsymbol{R}^{(2N+1)n}$; $\tilde{\boldsymbol{Z}} = [\tilde{\boldsymbol{x}}_{\text{non}}^{\mathrm{T}}, \tilde{\boldsymbol{\lambda}}_{\text{non}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbf{R}^{(2N+2)n}$; $\tilde{V} \in \mathbf{R}^{(2N+1)n\times(2N+2)n}$.

Since the initial condition of state variable is known as shown in Eq. (45), Eq. (48) can be rewritten as

$$\begin{bmatrix} \tilde{\boldsymbol{E}} & \tilde{\boldsymbol{F}} \\ \tilde{\boldsymbol{G}} & \tilde{\boldsymbol{H}} \\ \tilde{\boldsymbol{P}}_1 & \tilde{\boldsymbol{P}}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_{\text{non}} \\ \tilde{\boldsymbol{\lambda}}_{\text{non}} \end{bmatrix} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{Z}} = \tilde{\boldsymbol{V}}_1\tilde{\boldsymbol{x}}_1 + \tilde{\boldsymbol{V}}_{\text{e}}\tilde{\boldsymbol{Z}}_{\text{e}} = \boldsymbol{T} \tag{52}$$

where $\tilde{\boldsymbol{Z}}_{\text{e}} = [\tilde{\boldsymbol{x}}_2^{\mathrm{T}}, \tilde{\boldsymbol{x}}_3^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{x}}_{N+1}^{\mathrm{T}}, \tilde{\boldsymbol{\lambda}}_1^{\mathrm{T}}, \tilde{\boldsymbol{\lambda}}_2^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{\lambda}}_{N+1}^{\mathrm{T}}]^{\mathrm{T}} \in \boldsymbol{R}^{(2N+1)n}$ includes all the unknown variables; $\tilde{\boldsymbol{V}}_1 = [[\tilde{\boldsymbol{E}}]_{k1}^{\mathrm{T}}, [\tilde{\boldsymbol{G}}]_{k1}^{\mathrm{T}}, \boldsymbol{0}_{n\times n}]^{\mathrm{T}} \in \mathbf{R}^{(2N+1)n\times n}$ is a submatrix of $\tilde{\boldsymbol{V}}$; $\tilde{\boldsymbol{V}}_{\text{e}} \in \mathbf{R}^{(2N+1)n\times(2N+1)n}$ includes the rest parts of $\tilde{\boldsymbol{V}}$. Then, the well-posed linear algebraic equations expressed as Eq. (52) is solved for obtaining $\tilde{\boldsymbol{Z}}_{\text{e}}$ as

$$\begin{aligned} \tilde{\boldsymbol{Z}}_{\text{e}} &= \tilde{\boldsymbol{V}}_{\text{e}}^{-1}(\boldsymbol{T} - \tilde{\boldsymbol{V}}_1\tilde{\boldsymbol{x}}_1) = \tilde{\boldsymbol{V}}_{\text{e}}^{-1}(\boldsymbol{T} - \tilde{\boldsymbol{V}}_1\tilde{\boldsymbol{x}}(t)) \\ &= \tilde{\boldsymbol{V}}_{\text{e}}^{-1}(\boldsymbol{T} - \tilde{\boldsymbol{V}}_1\tilde{\boldsymbol{x}}_0) \end{aligned} \tag{53}$$

and $\tilde{\boldsymbol{Z}}$ can be rewritten as

$$\begin{aligned} \tilde{\boldsymbol{Z}} &= \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{Z}}_{\text{e}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n\times n} \\ -\tilde{\boldsymbol{V}}_{\text{e}}^{-1}\tilde{\boldsymbol{V}}_1 \end{bmatrix}\tilde{\boldsymbol{x}}_0 + \begin{bmatrix} \boldsymbol{0}_{n\times n} \\ \tilde{\boldsymbol{V}}_{\text{e}}^{-1}\boldsymbol{T} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I}_{n\times n} \\ \tilde{\boldsymbol{W}}_1 \end{bmatrix}\tilde{\boldsymbol{x}}_0 + \begin{bmatrix} \boldsymbol{0}_{n\times n} \\ \tilde{\boldsymbol{W}}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{W}}_{1\_x} \\ \tilde{\boldsymbol{W}}_{1\_\lambda} \end{bmatrix}\tilde{\boldsymbol{x}}_0 + \begin{bmatrix} \tilde{\boldsymbol{W}}_{2\_x} \\ \tilde{\boldsymbol{W}}_{2\_\lambda} \end{bmatrix} \end{aligned} \tag{54}$$

where $\tilde{\boldsymbol{W}}_{1\_x}, \tilde{\boldsymbol{W}}_{1\_\lambda} \in \mathbf{R}^{(N+1)n\times n}$ are partitions of the matrix $[\boldsymbol{I}_{n\times n}, \quad \tilde{\boldsymbol{W}}_1^{\mathrm{T}}]^{\mathrm{T}}$; $\tilde{\boldsymbol{W}}_{2\_x}, \tilde{\boldsymbol{W}}_{2\_\lambda} \in \mathbf{R}^{(N+1)n\times n}$ are partitions of the matrix $[\boldsymbol{0}_{n\times n}, \quad \tilde{\boldsymbol{W}}_2^{\mathrm{T}}]^{\mathrm{T}}$, so that $\tilde{\boldsymbol{x}}_j$ and $\tilde{\boldsymbol{\lambda}}_j$ are given by

$$\begin{cases} \tilde{\boldsymbol{x}}_j = (\tilde{\boldsymbol{W}}_{1\_x})_j\tilde{\boldsymbol{x}}_0 + (\tilde{\boldsymbol{W}}_{2\_x})_j \\ \tilde{\boldsymbol{\lambda}}_j = (\tilde{\boldsymbol{W}}_{1\_\lambda})_j\tilde{\boldsymbol{x}}_0 + (\tilde{\boldsymbol{W}}_{2\_\lambda})_j \end{cases} \tag{55}$$

where $(\tilde{\boldsymbol{W}}_{1\_x})_j, (\tilde{\boldsymbol{W}}_{1\_\lambda})_j \in \mathbf{R}^{n\times n}$ are partitions of $\tilde{\boldsymbol{W}}_{1\_x}$ and $\tilde{\boldsymbol{W}}_{1\_\lambda}$, respectively; $(\tilde{\boldsymbol{W}}_{2\_x})_j, (\tilde{\boldsymbol{W}}_{2\_\lambda})_j \in \mathbf{R}^{n\times n}$ are partitions of $\tilde{\boldsymbol{W}}_{2\_x}$ and $\tilde{\boldsymbol{W}}_{2\_\lambda}$, respectively. Thus, by substituting Eq. (55) into Eq. (44), the discretization for the optimal control law is obtained as

$$\begin{aligned} \tilde{\boldsymbol{u}}_j &= (\boldsymbol{u}_{\text{d}})_j - \boldsymbol{N}_j^{-1}\underset{\sim}{\hat{\boldsymbol{B}}}_j^{\mathrm{T}}\tilde{\boldsymbol{\lambda}}_j \\ &= (\boldsymbol{u}_{\text{d}})_j - \boldsymbol{N}_j^{-1}\underset{\sim}{\hat{\boldsymbol{B}}}_j^{\mathrm{T}}((\tilde{\boldsymbol{W}}_{1\_\lambda})_j\tilde{\boldsymbol{x}}_0 + (\tilde{\boldsymbol{W}}_{1\_\lambda})_j) \end{aligned} \tag{56}$$

The procedure for implementing the RHC for nonlinear problem similar to that in Ref.[17] is summarized as follows:

Step 1. For the given horizon length T and the desired trajectory $(\boldsymbol{x}_{\text{d}}, \boldsymbol{u}_{\text{d}})$, guess a nominal trajectory for the interval $\tau \in [t, t + T]$ and apply the quasi linearization method to the RHC problem for nonlinear system. A suitable guess would be the desired trajectory.

Step 2. Solve the sequence of TPBVPs:
- (a) Solve the linear equations Eq. (52).
- (b) Update the nominal trajectory using Eqs. (55) and (56).
- (c) Repeat steps (a) and (b) using the solution from the previous iteration as initial guess until the 2-norm of the change in state variable $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|$ is within a prescribed tolerance ε or a given iteration number is exceeded.

Step 3. Apply the optimal control input at the current time which is obtained from the converged trajectory from Step. 2 to the nonlinear system, and repeat Steps 1 and 2 until $\tau = \tau_{\text{f}}$ (τ_{f} is the end time).

As mentioned above, as the quasi linearization method is employed, in order to get the optimal control at any instantaneous time $t \geqslant \tau_0$, the resulting TPBVPs need to solved repeatedly at the current time until the state deviations between the current iteration and the previous iteration are within a prescribed tolerance or a given iteration number is exceeded. However, as the linearization method is employed, the resulting TPBVP only needs to be solved once at the current time.

4. Applications

4.1. LTV system with an analytical model

The following two-dimension LTV system with an analytical model is considered:[15]

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 2\tau & 1 \\ 0 & \tau+1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ \dfrac{2\tau+2}{2\tau+3} \end{bmatrix} u \tag{57}$$

with the performance index and the terminal constraint shown as Eqs. (2) and (3), respectively. The preceding system is an open-loop unstable system, but it is controllable for all $\tau \geqslant 0$.

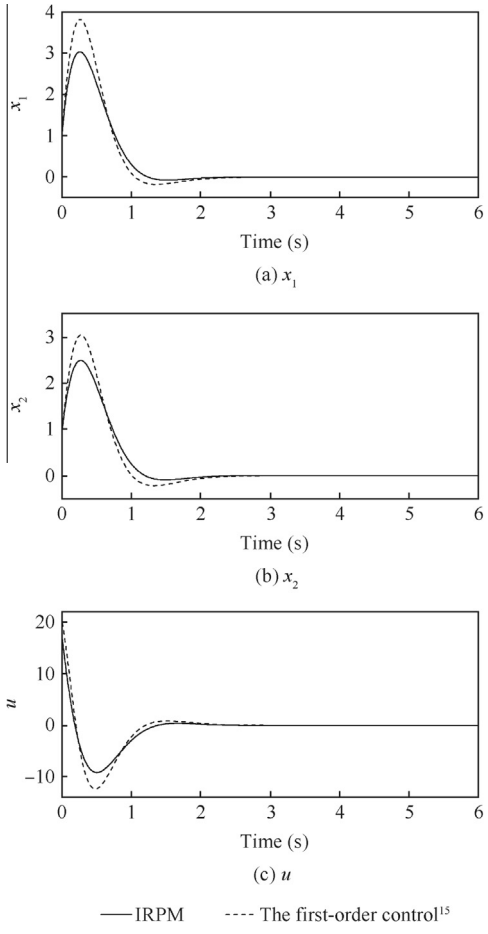

Fig. 1 Comparison between results of IRPM and the first-order control law.[15]

With the initial condition $\boldsymbol{x}_0 = [1, 1]^{\mathrm{T}}$, the weight matrices $\boldsymbol{Q} = \mathrm{diag}\{1, 1\}$ and $\boldsymbol{N} = 1$, the simulation step 0.01 s, the LGR points' number $N = 2$, and the horizon length $T = 0.6$ s, the preceding RHC problem is solved using the method proposed in Section 3.1. The results are shown in Fig. 1 alongside the results obtained from the first-order control law derived in Ref.[15]. As the first-order control law is derived based on the subinterval number $\bar{N} = 2$ and the subinterval length $h = 0.3$ s, the horizon length $T = \bar{N}h = 0.6$ s equals that was used in the IRPM. The responses as shown in Fig. 1 clearly demonstrate that the closed-loop system is stable in both cases, while the results of the IRPM have a smaller control input peak value and a smaller overshoot. Thus, with the same horizon length and the same subinterval number, the IRPM has a better performance than the first-order control law.[15]

Next, simulations are made with various LGR points' numbers and horizon lengths to learn more about the effects on the results.

The computation time for the optimal control input at the current time in all cases are shown in Table 1. It is observed that all the maximum computation time $t_{\mathrm{cal\_max}}$ and the average computation time $t_{\mathrm{cal\_ave}}$ are much shorter than the specified simulation step 0.01 s, demonstrating high computation efficiency of the proposed method. Two key related features are also seen from Table 1. First, as the LGR points' number N increases while the horizon length is fixed, the average computation time $t_{\mathrm{cal\_ave}}$ increases. Second, as the horizon length increases while the LGR points' number N is fixed, there are little differences between the average computation time $t_{\mathrm{cal\_ave}}$ in all cases. This is because that the most time-consuming work in the IRPM is the inverse of $\boldsymbol{V}_{\mathrm{e}}$ as shown in Eq. (28), and increasing the LGR points' number leads to increase of the dimension of $\boldsymbol{V}_{\mathrm{e}}$ which results in a longer average computation time, while increasing the horizon length has no effect on the dimension of $\boldsymbol{V}_{\mathrm{e}}$.

Table 1 Computation time in all cases.

N	T (s)	$t_{\mathrm{cal\_max}}$ (10^{-3} s)	$t_{\mathrm{cal\_ave}}$ (10^{-4} s)
2	0.6	3.3	2.0436
4	0.6	5.8	4.2122
6	0.6	3.3	7.9902
8	0.6	3.8	9.5169
10	0.6	4.6	14.0000
2	0.3	4.5	1.9171
2	0.9	3.2	1.9975
2	1.2	3.0	1.8979
2	1.5	2.7	1.7453

Fig. 2(a)–2(c) give the results for the LGR points' number $N = 2$ to $N = 10$ by steps of 2 with the horizon length $T = 0.6$ s. It is observed that there are only small differences between the results for cases of different LGR points' numbers with the same horizon length. Regarding the result for $N = 10$ as the reference, it can be seen that state and control curves of other cases converge to the reference curves (x_{1_ref}, x_{2_ref}, u_{ref}) with great rapidity. The maximum absolute errors between the reference result and another result are defined as

$$\begin{cases} e_{x_1} = \max(\lg|x_1 - x_{1\_\mathrm{ref}}|) \\ e_{x_2} = \max(\lg|x_2 - x_{2\_\mathrm{ref}}|) \\ e_u = \max(\lg|u - u_{\mathrm{ref}}|) \end{cases} \tag{58}$$

The values of e_{x_1}, e_{x_2}, e_u are shown in Fig. 2(d). It can be seen that all errors decrease nearly over a linear trend until $N = 8$, demonstrating the spectral convergence rate of the Radau pseudospectral approximation.

Fig. 3 gives the results for horizon length $T = 0.3$ s to $T = 1.5$ s by steps of 0.3 s with LGR points' number $N = 2$. It is observed that there are large differences between the results for cases of different horizon lengths with the same LGR points' number. For smaller horizon length which can be viewed as achieving the control objective in a faster way, the control input peak value and the overshoot are larger while the stabilization time is shorter.

Thus, it can be concluded that the computation efficiency of the optimal control law depends on the LGR points' number,

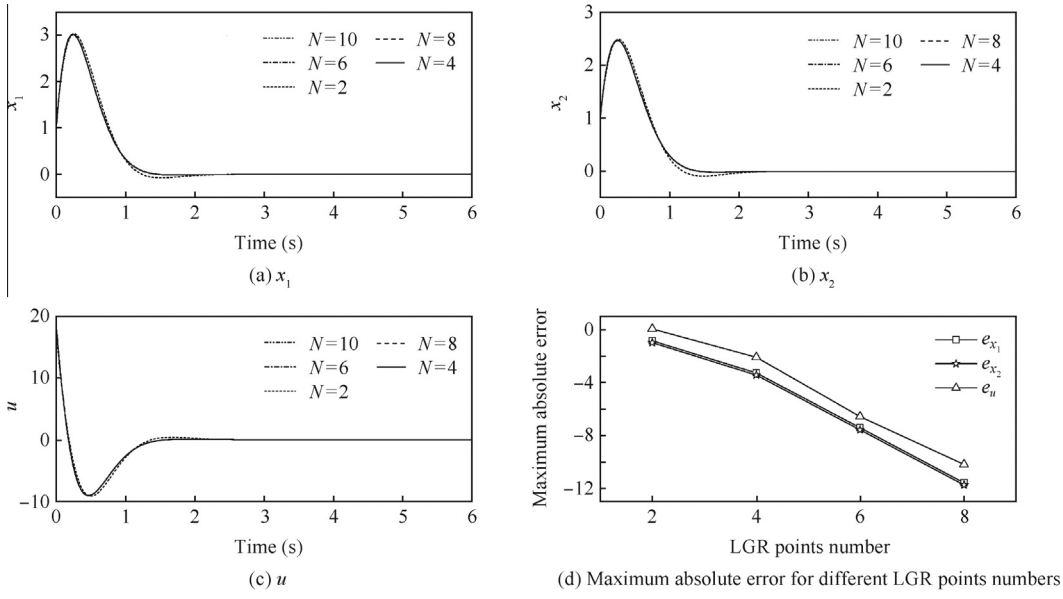

Fig. 2 Comparisons between results for cases of different LGR points' numbers with horizon length $T = 0.6$ s.

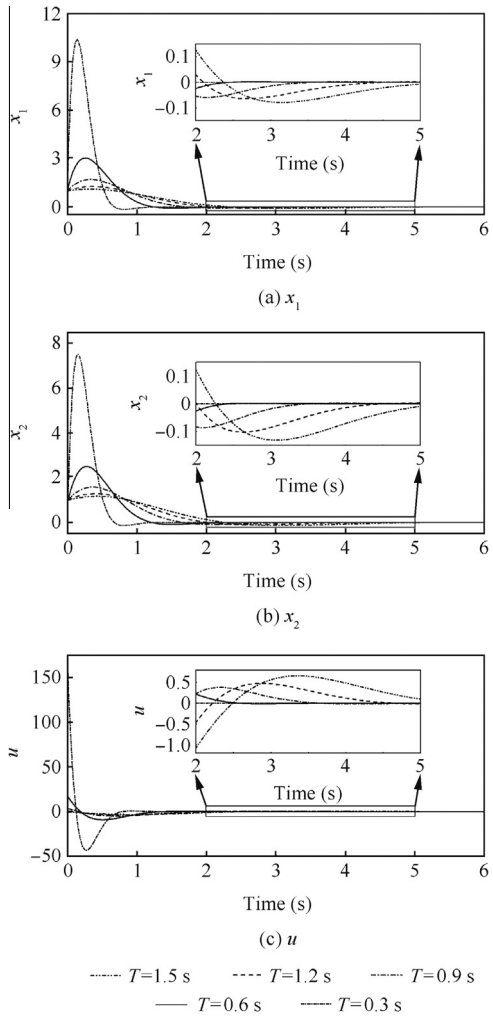

Fig. 3 Comparisons between results for cases of different horizon length with LGR points' number $N = 2$.

while the gains of the optimal control law are mainly affected by the horizon length.

4.2. Single inverted pendulum

The single inverted pendulum depicted in Fig. 4 is a complex, nonlinear, coupling, and open-loop unstable system with equations of motion expressed as follows:

$$\begin{cases} (M+m)\ddot{x} + b_2\dot{x} - ml\ddot{\phi}\cos\phi + ml\dot{\phi}^2\sin\phi = u \\ (I+ml^2)\ddot{\phi} + b_1\dot{\phi} - ml\ddot{x}\cos\phi - mgl\sin\phi = 0 \end{cases} \tag{59}$$

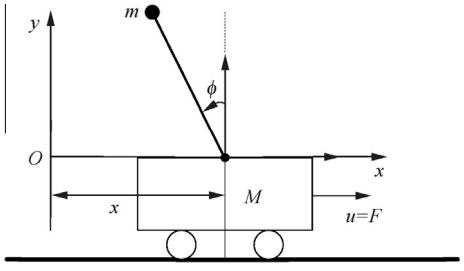

Fig. 4 Single inverted pendulum.

where $M = 0.5$ kg and $m = 0.2$ kg are mass of the driving cart and the pole, respectively; x is the position of the driving cart; $l = 0.3$ m is the length of the pole; ϕ is the inclined angle between the pole and the vertical upward direction measured counterclockwise for the vertical upward direction; F is the traction on the driving cart which is the control variable u of the inverted pendulum; $I = ml^2/3$ is the rotational inertia of the pole; $b_1 = 0.001\ \mathrm{N\cdot m\cdot s/rad}$ is the friction coefficient between the pole and the driving cart; $b_2 = 0.1\ \mathrm{N\cdot s/m}$ is the friction coefficient between the driving cart and the horizontal plane. The performance index is shown as Eq. (8).

The control task is set as $\boldsymbol{x}_0 = [x, \dot{x}, \phi, \dot{\phi}]^{\mathrm{T}}|_{\tau=0} = [0.5\ \mathrm{m}, 0\ \mathrm{m/s}, -0.1\ \mathrm{rad}, 0\ \mathrm{rad/s}]^{\mathrm{T}}$ and $\boldsymbol{x}_{\mathrm{f}} = \boldsymbol{\psi} = [0\ \mathrm{m}, 0\ \mathrm{m/s}, 0\ \mathrm{rad}, 0\ \mathrm{rad/s}]^{\mathrm{T}}$ with the weight matrices $\boldsymbol{Q} = \mathrm{diag}\{100, 0, 40, 0\}$, $\boldsymbol{N} = 1$, $\boldsymbol{S}_{\mathrm{f}} = \boldsymbol{M}_{\mathrm{f}} = \boldsymbol{I}_{4\times4}$, and the simulation step 0.01 s. The desired state is set as $\boldsymbol{x}_{\mathrm{d}} = \boldsymbol{x}_{\mathrm{f}}$ and the desired control is set as $\boldsymbol{u}_{\mathrm{d}} = \boldsymbol{0}$. The preceding RHC problem is solved using the method proposed in Section 3.2 with the LGR points' number $N = 10$ and the horizon length $T = 2$ s. In this example, we have ignored the computation time delay between solving the RHC problem and the implementation of the control since we mainly focus on the performance of the IRPM and the corresponding controller.

Fig. 5 gives the results of the linearization method and the quasi linearization method with the tolerance $\varepsilon = 1 \times 10^{-6}$ and the maximum iteration number $n = 3$. $E_\phi, E_{\dot{\phi}}, E_x, E_{\dot{x}}$ and E_u in Fig. 5 denote the errors for corresponding variables between the results of the linearization method and the results of the quasi linearization method. It can be seen that there are small differences between the results of both methods and errors of the results converge to 0 rapidly. Both of the methods have achieved the control objective and both of the results demonstrate the closed-loop stability of controlling the preceding system using the IRPM.

Since the RHC problem for nonlinear system is approximated with successive linear approximations via the linearization method or the quasi linearization method, the effects caused by the LGR points' number and the horizon length on the results would be similar to that of the RHC problem for LTV system. Simulations are made with various maximum iteration numbers and tolerances to further study the effects on the results with the quasi linearization method. Results show that the control objective has been achieved in all simulation cases.

First, results of cases for various tolerances without limit of maximum iteration number are compared. We choose the result of the case $\varepsilon = 1 \times 10^{-10}$ as the reference result. The errors of the performance index, state variables, costate variables, and control variables between the reference result and the result of another case are defined as

$$\begin{cases} \hat{e}_J = |J - J^*| \\ \hat{e}_{\boldsymbol{x}} = \max(|\phi - \phi^*|, |\dot{\phi} - \dot{\phi}^*|, |x - x^*|, |\dot{x} - \dot{x}^*|) \\ \hat{e}_{\boldsymbol{\lambda}} = \max(|\lambda_\phi - \lambda_\phi^*|, |\lambda_{\dot{\phi}} - \lambda_{\dot{\phi}}^*|, |\lambda_x - \lambda_x^*|, |\lambda_{\dot{x}} - \lambda_{\dot{x}}^*|) \\ \hat{e}_u = \max(|u - u^*|) \end{cases} \tag{60}$$

where the symbol * represents the reference result. Table 2 shows the errors of cases for various tolerances. It is observed that the errors of all cases are very small and results are closer to the reference result as the tolerance ε decreases. It is worth noticing that with the prescribed tolerance, more iteration

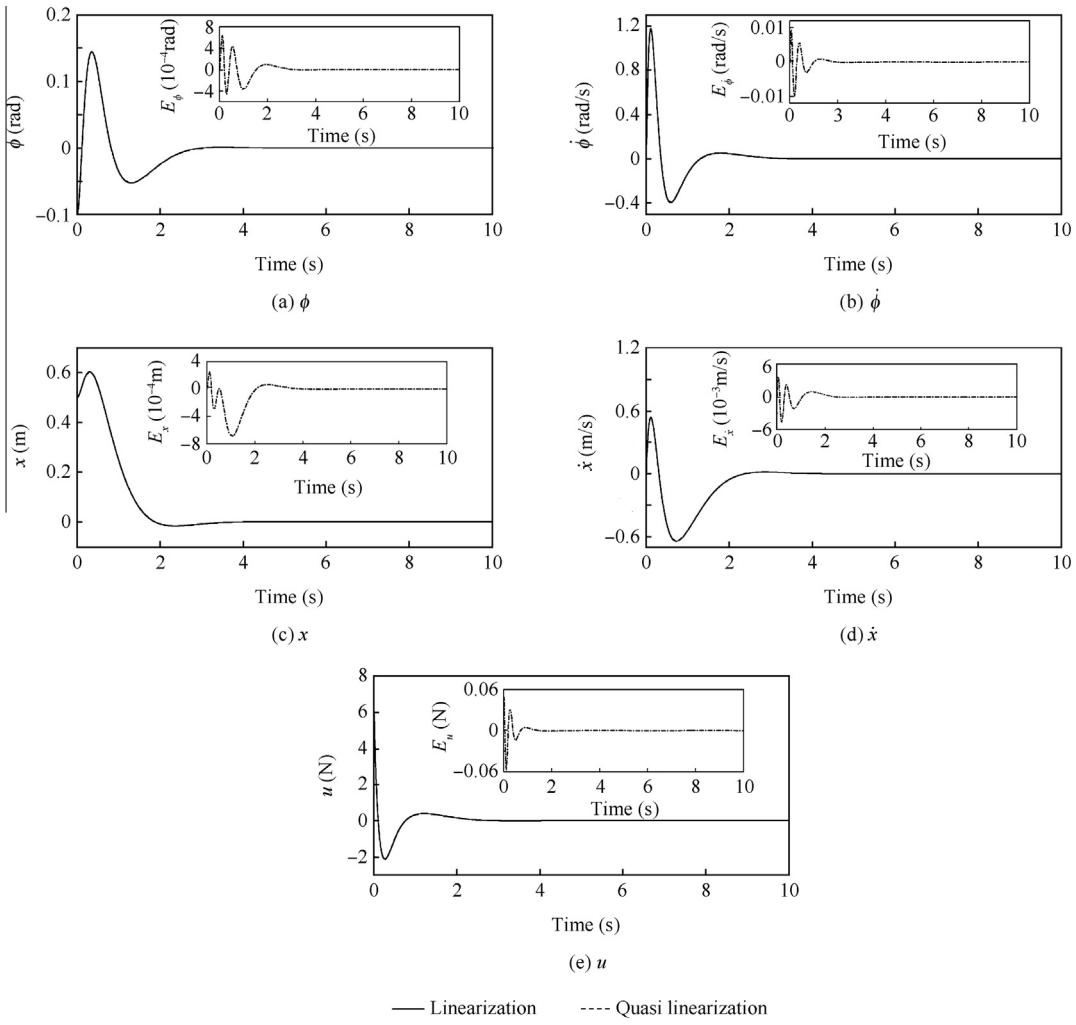

Fig. 5 Comparison between results of linearization method and quasi linearization method.

numbers are needed at the beginning of the simulation while iteration numbers decrease as time goes on. Table 2 demonstrates the high accuracy of the IPRM.

Then, results of cases for various maximum iteration numbers without limit of tolerance are compared. The errors are defined as

$$\begin{cases} \tilde{e}_J = |J^k - J^{k-1}| \\ \tilde{e}_{\boldsymbol{x}} = \max(|\phi^k - \phi^{k-1}|, |\dot{\phi}^k - \dot{\phi}^{k-1}|, |x^k - x^{k-1}|, |\dot{x}^k - \dot{x}^{k-1}|) \\ \tilde{e}_{\boldsymbol{\lambda}} = \max(|\lambda_\phi^k - \lambda_\phi^{k-1}|, |\lambda_{\dot{\phi}}^k - \lambda_{\dot{\phi}}^{k-1}|, |\lambda_x^k - \lambda_x^{k-1}|, |\lambda_{\dot{x}}^k - \lambda_{\dot{x}}^{k-1}|) \\ \tilde{e}_u = \max(|u^k - u^{k-1}|) \end{cases} \tag{61}$$

where the symbol k represents the result of maximum iteration number $n = k$. Table 3 shows the errors of the cases for various maximum iteration numbers. It is observed that the errors decrease fast as the maximum iteration number increases before the maximum iteration number reaches 7, and then the errors remain relatively unchanged. Table 3 demonstrates the fast convergence speed of the IRPM.

It is evident that both the decrease of the tolerance and the increase of maximum iteration number require more computation time when using the quasi linearization method. Thus, we should properly choose the tolerance and the maximum itera-

Table 2 Errors between reference results and results of other cases with various tolerances.

ε	$\hat{e}_J$	$\hat{e}_{\boldsymbol{x}}$	$\hat{e}_{\boldsymbol{\lambda}}$	$\hat{e}_u$
1	2.36×10^{-4}	4.49×10^{-3}	1.06×10^{-1}	1.50×10^{-2}
1×10^{2}	2.82×10^{-6}	8.90×10^{-5}	2.46×10^{-3}	1.01×10^{-3}
1×10^{-4}	6.62×10^{-8}	1.01×10^{-6}	1.20×10^{-5}	9.23×10^{-6}
1×10^{-6}	8.03×10^{-10}	7.02×10^{-9}	1.22×10^{-7}	7.81×10^{-8}
1×10^{-8}	2.43×10^{-11}	8.05×10^{-11}	4.55×10^{-9}	9.03×10^{-10}

Table 3 Errors between result of maximum iteration number $n = k$ and results of maximum iteration number $n = k - 1$.

k	$\tilde{e}_J$	$\tilde{e}_x$	$\tilde{e}_\lambda$	$\tilde{e}_u$
1	1.66×10^{-4}	1.01×10^{-2}	4.48×10^{-1}	5.68×10^{-2}
3	4.24×10^{-7}	4.23×10^{-6}	6.68×10^{-5}	1.60×10^{-5}
5	1.93×10^{-10}	6.36×10^{-9}	1.14×10^{-7}	3.33×10^{-8}
7	3.14×10^{-11}	5.12×10^{-11}	4.16×10^{-9}	1.07×10^{-9}
9	1.79×10^{-11}	4.81×10^{-11}	4.49×10^{-9}	1.10×10^{-9}
11	1.00×10^{-11}	9.16×10^{-11}	4.16×10^{-9}	1.21×10^{-9}

tion number to ensure the computation efficiency and meet the requirement of accuracy at the same time.

4.3. Space shuttle reentry guidance

The space shuttle reentry guidance problem[3,7] is considered in this section. The equations of translational motion for the space shuttle over a spherical, non-rotating earth are given by[3,7]

$$\begin{cases} \dot{r} = V\sin\gamma \\ \dot{\theta} = \dfrac{V\cos\gamma\sin\psi}{r\cos\varphi} \\ \dot{\varphi} = \dfrac{V\cos\gamma\cos\psi}{r} \\ \dot{V} = -\dfrac{D}{m_1} - g\sin\gamma \\ \dot{\gamma} = \dfrac{1}{V}\left[\dfrac{L\cos\sigma}{m_1} + \left(\dfrac{V^2}{r} - g\right)\cos\gamma\right] \\ \dot{\psi} = \dfrac{1}{V}\left(\dfrac{L\sin\sigma}{m_1\cos\gamma} + \dfrac{V^2}{r}\cos\gamma\sin\psi\tan\varphi\right) \end{cases} \tag{62}$$

where r, θ, φ, V, γ and ψ are radial distance, longitude, latitude, velocity, flight-path angle and heading angle, respectively; $m_1 = 92709$ kg is the mass of the space shuttle; $g = \mu/r^2$ is the gravity acceleration and $\mu = 3.9805 \times 10^{14}$ m^3/s^2 is the geocentric gravitational constant; σ is the bank angle; L and D are lift force and drag force, respectively, and are expressed as

$$L = C_L qS, \quad D = C_D qS \tag{63}$$

where $S = 249.9092$ m^2 is the reference area; $q = 0.5\rho V^2$ is the dynamic pressure; ρ is the atmosphere density and is based on the exponential model $\rho = \rho_0 e^{-(r-R_0)/H}$; $\rho_0 = 1.225$ kg/m^3 is the sea level atmosphere density; $R_0 = 6371004$ m is the earth radius; $H = 7254.24$ m is the density scale height; the lift coefficient C_L and the drag coefficient C_D are given as

$$\begin{cases} C_L = C_{L0} + C_{L\alpha}\alpha \\ C_D = C_{D0} + C_{D\alpha}\alpha + C_{D\alpha^2}\alpha^2 \end{cases} \tag{64}$$

where $C_{L0} = -0.207$, $C_{L\alpha} = 1.6756$, $C_{D0} = 0.0785$, $C_{D\alpha} = -0.3529$, $C_{D\alpha^2} = 2.04$, α is the angle of attack measured in radians. The angle of attack α and the bank angle σ are the trajectory control inputs.

The path constraints consisting of the heating rate constraint, the dynamic pressure constraint and the total load constraint are expressed as

$$\dot{Q} = Q_\alpha Q_r \leqslant \dot{Q}_{\max} = 2 \times 10^6 \text{ W/m}^2 \tag{65}$$

$$q = 0.5\rho V^2 \leqslant q_{\max} = 2 \times 10^4 \text{ Pa} \tag{66}$$

$$n = \sqrt{L^2 + D^2}/(m_1 g) \leqslant n_{\max} = 2.5 \tag{67}$$

where $Q_\alpha = 1.067 - 1.101\alpha + 0.6988\alpha^2 - 0.1903\alpha^3$, $Q_r = 1.7822 \times 10^{-4}\rho^{0.5}V^{3.07}$. The constraints on the control inputs are given as

$$\begin{cases} \alpha \in [0^\circ, 40^\circ], & \dot{\alpha} \in [-5(^\circ)/\text{s}, 5(^\circ)/\text{s}] \\ \sigma \in [-85^\circ, 85^\circ], & \dot{\sigma} \in [-20(^\circ)/\text{s}, 20(^\circ)/\text{s}] \end{cases} \tag{68}$$

The initial state constraints are $h_0 = 79250$ m, $\theta_0 = 0^\circ$, $\varphi_0 = 0^\circ$, $V_0 = 7800$ m/s, $\gamma_0 = -1^\circ$ and $\psi_0 = 90^\circ$, and the terminal state constraints are $h_f = 24380$ m, $\theta_f = 80^\circ$, $\varphi_f = 30^\circ$, $V_f = 762$ m/s and $\gamma_f = -5^\circ$.

The desired trajectory is generated using the open source software GPOPS.[24] In GPOPS, the Radau pseudospectral method[19–22] is employed to discretize the trajectory optimization problem and the resulting nonlinear programming problem is solved by the limit version of the software SNOPT.[25] The LGR points' number is $N = 100$ and the object function expressed as follows is to minimize the maximum heating rate of the whole trajectory

$$J_{\text{ref}} = \min(\max(\dot{Q}(t))) \tag{69}$$

The time histories of the desired trajectory, the corresponding trajectory control inputs and the path constraints are shown in Figs. 6 and 7 by solid lines, where it can be seen that all constraints are satisfied. The flight time is 2047.59 s and the maximum heating rate of the whole trajectory is 7.6452×10^5 W/m^2.

With the desired trajectory and subject to the nonlinear equations of motion shown as Eq. (62), the space shuttle reentry guidance problem, which is regarded as a trajectory tracking problem, can be formulated as an RHC problem for nonlinear system and is solved through the proposed method in Section 3.2. The approximation method used in this example is the linearization method. The nonzero elements of the Jacobian matrices $\hat{A}(\tau) \in \mathbf{R}^{6\times6}$ and $\hat{B}(\tau) \in \mathbf{R}^{6\times3}$ in Eq. (33) can be analytically expressed as $\hat{A}_{14} = \sin\gamma$, $\hat{A}_{15} = V\cos\gamma$, $\hat{A}_{21} = -\dfrac{V\cos\gamma\sin\psi}{r^2\cos\theta}$, $\hat{A}_{23} = \dfrac{V\cos\gamma\sin\psi\tan\theta}{r\cos\theta}$, $\hat{A}_{24} = \dfrac{\cos\gamma\sin\psi}{r\cos\theta}$, $\hat{A}_{25} = -\dfrac{V\sin\gamma\sin\psi}{r\cos\theta}$, $\hat{A}_{26} = \dfrac{V\cos\gamma\cos\psi}{r\cos\theta}$, $\hat{A}_{31} = -\dfrac{V\cos\gamma\cos\psi}{r^2}$, $\hat{A}_{34} = \dfrac{\cos\gamma\cos\psi}{r}$, $\hat{A}_{35} = -\dfrac{V\sin\gamma\cos\psi}{r}$, $\hat{A}_{36} = -\dfrac{V\cos\gamma\sin\psi}{r}$, $\hat{A}_{41} = \dfrac{KC_D V^2}{H} + \dfrac{2g\sin\gamma}{r}$, $\hat{A}_{44} = -2KC_D V$, $\hat{A}_{45} = -g\cos\gamma$, $\hat{A}_{51} = -\dfrac{KC_L V\cos\sigma}{H} - \dfrac{V\cos\gamma}{r^2} + \dfrac{2g\cos\gamma}{Vr}$, $\hat{A}_{54} = KC_L\cos\sigma + \left(\dfrac{g}{V^2} + \dfrac{1}{r}\right)\cos\gamma$, $\hat{A}_{55} = \left(\dfrac{g}{V} + \dfrac{V}{r}\right)\sin\gamma$, $\hat{A}_{61} = -\dfrac{KC_L V\sin\sigma}{H\cos\gamma} - \dfrac{V\cos\gamma\sin\psi\tan\theta}{r^2}$, $\hat{A}_{63} = \dfrac{V\cos\gamma\sin\psi(1+\tan^2\theta)}{r}$, $\hat{A}_{64} = \dfrac{KC_L\sin\sigma}{\cos\gamma} + \dfrac{\cos\gamma\sin\psi\tan\theta}{r}$, $\hat{A}_{65} = \dfrac{KC_L V\sin\sigma\tan\gamma}{\cos\gamma} - \dfrac{V\sin\gamma\sin\psi\tan\theta}{r}$, $\hat{A}_{66} = \dfrac{V\cos\gamma\cos\psi\tan\theta}{r}$, $\hat{B}_{41} = -kV^2(C_{D\alpha} + 2C_{D\alpha^2}\alpha)$, $\hat{B}_{51} = KVC_{L\alpha}\cos\sigma$, $\hat{B}_{52} = -KVC_L\sin\sigma$, $\hat{B}_{61} = KVC_{L\alpha}\sin\sigma$, $\hat{B}_{62} = KVC_L\cos\sigma$ with $K = 0.5\,\rho S/m_1$, and are valued based

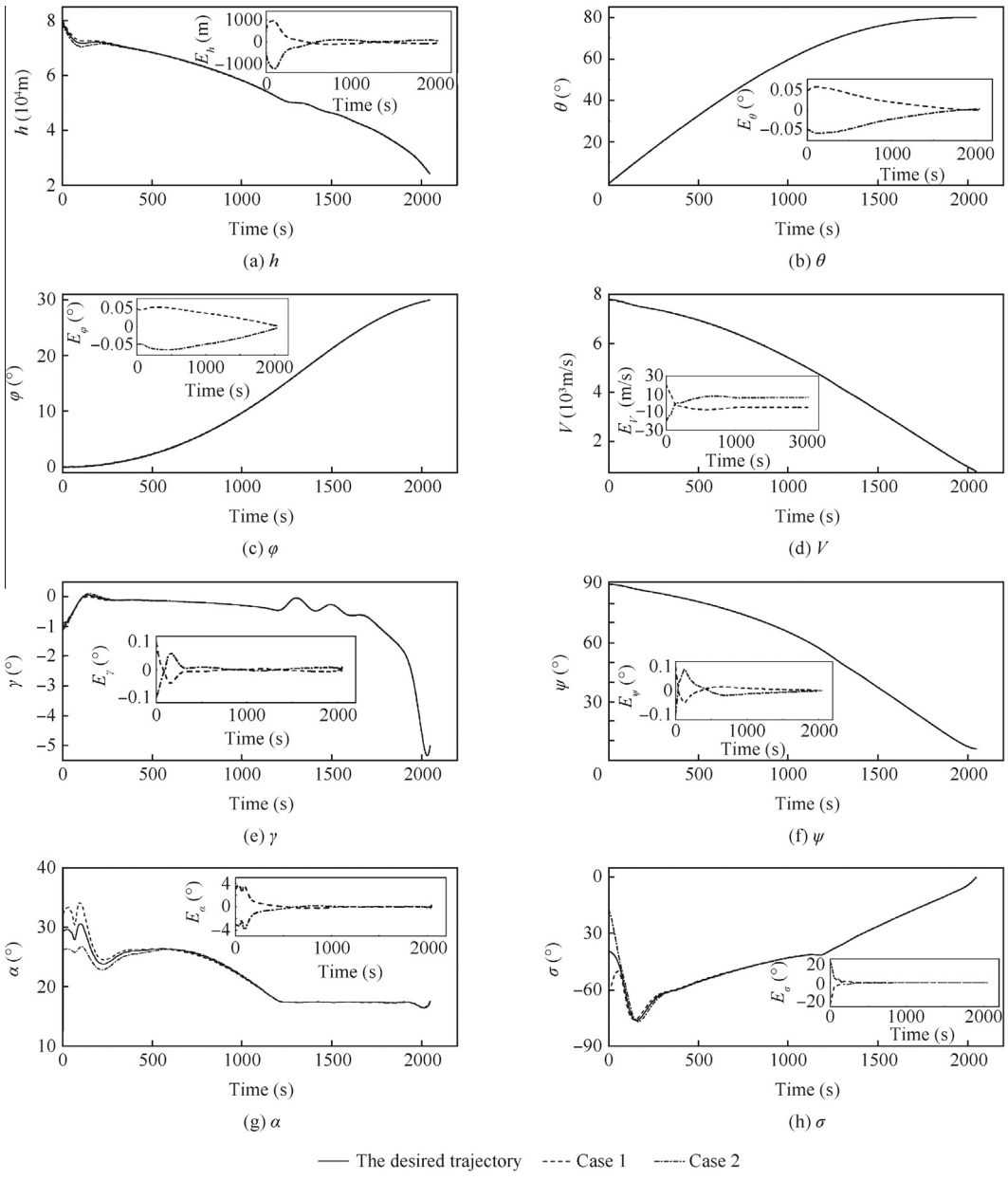

Fig. 6 Time histories of state variables and control variables.

on the time-varying desired trajectory so that the Jacobian matrices are pure functions of time.

In the closed-loop guidance simulation, both the simulation step and the guidance cycle are set as 1s and the weight matrices are valued as $\boldsymbol{Q} = \mathrm{diag}\{7 \times 10^{-7}, 1.3 \times 10^{6}, 5.8 \times 10^{5}, 2.5 \times 10^{-3}, 1.5 \times 10^{5}, 1.5 \times 10^{5}\}$, $\boldsymbol{N} = \mathrm{diag}\{130, 8\}$, $\boldsymbol{S}_{\mathrm{f}} = \boldsymbol{0}$ with the horizon length $T = 200$ s and the LGR points' number for the IRPM $N = 20$. Two test cases with the initial state deviations set as $\Delta h_0 = 600$ m, $\Delta\theta_0 = 0.05°$, $\Delta\varphi_0 = 0.05°$, $\Delta V_0 = 20$ m/s, $\Delta\gamma_0 = 0.1°$, $\Delta\psi_0 = 0.1°$ for case 1 and $\Delta h_0 = -600$ m, $\Delta\theta_0 = -0.05°$, $\Delta\varphi_0 = -0.05°$, $\Delta V_0 = -20$ m/s, $\Delta\gamma_0 = -0.1°$, $\Delta\psi_0 = -0.1°$ for case 2 are simulated. Table 4 shows the computation time for the optimal guidance law and the errors of terminal states between the closed-loop trajectories and the desired trajectory. It is observed that the maximum computation time $t_{\mathrm{cal\_max}}$ and the average computation time $t_{\mathrm{cal\_ave}}$ for two cases are much shorter than the simulation step and the guidance cycle, which demonstrates that the optimal guidance law is obtained by the proposed IRPM with high computation efficiency. Fig. 6 gives the results of the closed-loop trajectories, the corresponding control inputs and their errors relative to the desired ones. $E_h, E_\theta, E_\varphi, E_V, E_\gamma, E_\psi, E_\alpha$ and E_σ in Fig. 6 denote the errors for corresponding variables between the results of the closed-loop trajectories and the results of the desired trajectory. It is observed that the closed-loop trajectory can track the desired trajectory with small terminal state errors in the presence of a rational range of initial state deviations. Fig. 7 shows the curves of path constraints of the closed-loop trajectories, where it can be seen that all path constraints are satisfied when

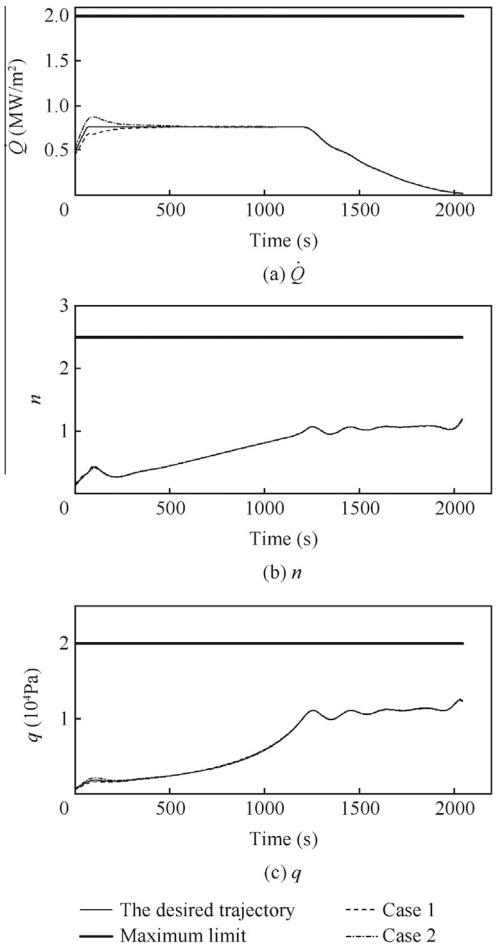

Fig. 7 Time histories of path constraints.

Table 4 Computation time for optimal guidance law and errors of terminal states between closed-loop trajectories and desired trajectory.

Variable	No. of cases	
	1	2
Δh_f (m)	−43.16	50.89
$\Delta\theta_f$ (°)	-1.64×10^{-3}	1.61×10^{-3}
$\Delta\varphi_f$ (°)	1.70×10^{-3}	-3.41×10^{-3}
ΔV_f (m/s)	−4.97	6.14
$\Delta\gamma_f$ (°)	2.39×10^{-3}	9.70×10^{-3}
$\Delta\psi_f$ (°)	-3.20×10^{-4}	-3.34×10^{-4}
t_{cal_max} (s)	0.15	0.13
t_{cal_ave} (s)	7.88×10^{-2}	7.09×10^{-2}

the guidance law is applied. From the simulation results, it can be concluded that the IRPM is effective to deal with the space shuttle reentry guidance problem.

5. Conclusions

(1) Using the IRPM, the TPBVP resulting from the RHC problem is discretized into well-posed linear algebraic equations which is solved efficiently and accurately via a matrix partitioning approach. Since complex computations such as backward sweep of the Riccati differential equation and transition matrices are avoided, the numerical stability and the high computation efficiency can be guaranteed.

(2) As the Radau pseudospectral approximation, employed in the IRPM, has an exponential convergence rate for smooth problems, highly accurate optimal solutions can be obtained via a small number of LGR points, which in turn can save computer memory storages and improve computation efficiency.

High computation efficiency and high accuracy of the IRPM and stability of the closed-loop systems with the resulting optimal feedback control law are validated through three numerical examples. Future work will concentrate on the application of the proposed method to other practical aerospace problems such as formation and station keeping of spacecraft, nanosatellite fast deorbit using electrodynamic tether, etc.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61174221 and 61402039).

References

1. Zhang XY, Duan HB, Yu YX. Receding horizon control for multi-UAVs close formation control based on differential evolution. *Sci China Inf Sci* 2010;**53**(2):223–35.
2. Zhang LM, Sun MW, Chen ZQ, Wang ZH, Wang YK. Receding horizon trajectory optimization with terminal impact specifications. *Math Prob Eng* 2014;604705-1-8.
3. Cai WW, Zhu YW, Yang LP, Zhang YW. Optimal guidance for hypersonic reentry using inversion and receding horizon control. *IET Control Theory Appl* 2015;**9**(9):1347–55.
4. Williams P. Spacecraft rendezvous on small relative inclination orbits using tethers. *J Spacecraft Rockets* 2005;**42**(6):1047–60.
5. Lu P. Regulation about time-varying trajectories: precision entry guidance illustrated. *J Guid Control Dyn* 1999;**22**(6):784–90.
6. Yang L, Zhou H, Chen WC. Application of linear Gauss pseudospectral method in model predictive control. *Acta Astronaut* 2014;**96**:175–87.
7. Tian BL, Zong Q. Optimal guidance for reentry vehicles based on indirect Legendre pseudospectral method. *Acta Astronaut* 2011;**68**(7–8):1176–84.
8. Peng HJ, Gao Q, Wu ZG, Zhong WX. Optimal guidance based on receding horizon control for low-thrust transfer to libration point orbits. *Adv Space Res* 2013;**51**(11):2093–111.
9. Park Y, Harmon CT. Autonomous real-time adaptive management of soil salinity using a receding horizon control algorithm: a pilot-scale demonstration. *J Environ Manage* 2011;**92**(10):2619–27.
10. Zavala VM, Biegler LT. Optimization-based strategies for the operation of low-density polyethylene tubular reactors: Nonlinear model predictive control. *Comput Chem Eng* 2009;**33**(10):1735–46.
11. Du Toit NE, Burdick JW. Robot motion planning in dynamics, uncertain environments. *IEEE Trans Rob* 2012;**28**(1):101–15.
12. Lee SM, Kim H, Myung H, Yao X. Cooperative coevolutionary algorithm-based model predictive control guaranteeing stability of multi-robot formation. *IEEE Trans Control Syst Technol* 2015;**23**(1):37–51.
13. Ohtsuka T, Fujii HA. Real-time optimization algorithm for nonlinear receding-horizon control. *Automatica* 1997;**33**(6):1147–54.

14. Bryson AE. *Dynamic optimization*. Menlo Park, CA: Addison-Wesley; 1999. p. 315–8.
15. Lu P. Closed-form control laws for linear time-varying systems. *IEEE Trans Autom Control* 2000;**45**(3):537–42.
16. Yan H, Fahroo F, Ross IM. Optimal feedback control laws by Legendre pseudospectral approximations. *Proceedings of the American control conference*; 2001 June 25–27; Arlington, VA. Piscataway (NJ), IEEE Press; 2001. p. 2388–93.
17. Williams P. Applications of pseudospectral methods for receding horizon control. *J Guid Control Dyn* 2004;**27**(2):310–4.
18. Peng HJ, Gao Q, Wu ZG, Zhong WX. Efficient sparse approach for solving receding-horizon control problems. *J Guid Control Dyn* 2013;**36**(6):1864–72.
19. Garg D, Patterson MA, Francolin C, Darby CL, Huntington GT, Hager WW, et al. Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method. *Comput Optim Appl* 2009;**49**(2):335–58.
20. Garg D, Hager WW, Rao AV. Pseudospectral methods for solving infinite-horizon optimal control problems. *Automatica* 2011;**47**(6):829–37.
21. Fahroo F, Ross IM. Pseudospectral methods for infinite-horizon optimal control problems. *J Guid Control Dyn* 2008;**31**(4):927–36.
22. Kameswaran S, Biegler LT. Convergence rates for direct transcription of optimal control problems using collocation at Radau points. *Comput Optim Appl* 2007;**41**(1):81–126.
23. Kwon WH, Pearson AE. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Trans Autom Control* 1977;**22**(5):838–42.
24. Rao AV, Benson DA, Darby CL, Mahon B, Francolin C, Patterson M, et al. User's manual for GPOPS version 4.x: A matlab software for solving multiple-phase optimal control problems using hp-adaptive pseudospectral methods. [Internet]. [cited 2011 August]. Available from: http://www.gpops2.com/usersManual/usersManual.html.
25. Gill PE, Murray W, Saunders MA. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev* 2002;**47**(1):99–131.

Yuxin Liao received the B.S. degree from Beihang University in 2010, and is currently a Ph.D. candidate at the same university. His area of research includes numerical optimal control, as well as hypersonic vehicle guidance.

Huifeng Li received the B.S. and Ph.D. degrees from Xi'an Jiaotong University in 1991 and 1998 respectively, and is currently a professor and Ph.D. supervisor at School of Astronautics, Beihang University. Her area of research includes hypersonic vehicle guidance and control, as well as dynamic modeling.